COMPONENT PART NOTICE

THIS PAPER IS A COMPONENT PART OF THE FOLLOWING COMPILATION REPORT:

TITLE: Computing Science and Statistics: Proceedings of the Symposium on Interface Critical Applications of Scientific Computing (23rd): Biology, Engineering, Medicine, Speech Held in Seattle, Washington on 21-24 April 1991.

TO ORDER THE COMPLETE COMPILATION REPORT, USE AD-A252 938.

THE COMPONENT PART IS PROVIDED HERE TO ALLOW USERS ACCESS TO INDIVIDUALLY AUTHORED SECTIONS OF PROCEEDING, ANNALS, SYMPOSIA, ETC. HOWEVER, THE COMPONENT SHOULD BE CONSIDERED WITHIN THE CONTEXT OF THE OVERALL COMPILATION REPORT AND NOT AS A STAND-ALONE TECHNICAL REPORT.

THE FOLLOWING COMPONENT PART NUMBERS COMPRISE THE COMPILATION REPORT:

AD#: AD-P007 096 thru AD#: AD-P007 225

AD#: AD#:

AD#: AD#:

Accesion For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

DTIC ELECTE JUL 23 1992 S A D

This document has been approved for public release and sale; its distribution is unlimited.

DTIC FORM 463 MAR 85

OPI: DTIC-TID

Parallel and Sequential Implementations for Combining Belief Functions*

Mary McLeish and Fei Song
Dept. of Computing and Information Science
University of Guelph
Guelph, Ontario, Canada N1G 2W1

Abstract

This paper reports our experiments about parallel and sequential implementations for combining belief functions with an application to a medical diagnostic system. We use as a basis existing methods for combining two belief functions: a direct combination based on Dempster's rule and an indirect combination through Möbius transforms. We further explore various parallel algorithms for combining more than two belief functions, as different belief functions can be combined in any order as long as they are independent of each other. Our results indicate that for the general case, the parallel implementation based on fast Möbius transforms proves to be the most efficient. However, for practical applications where most subsets of a frame of hypotheses have zero probabilities, the parallel implementation based on an improved direct combination rule remains the most efficient.

1 Introduction

This paper presents parallel and sequential algorithms for combining belief functions. The *Belief Function* approach for approximate reasoning, also called the Dempster-Shafer theory [Shafer, 1976], can be seen as a generalization of the *Probability* approach [Pearl, 1988], since probabilities are assigned directly to subsets of a set of mutually exclusive and exhaustive hypotheses rather than each of the hypotheses.

One important problem for the application of the DS-theory is the efficiency for combining the belief functions from different evidences. Barnett [1981] proposed a polynomial algorithm which only applies to sets of single hypotheses or singletons. Work by ([Shafer and Logan, 1987] and [Shafer *et al.*, 1987]) deals with extended subsets that form a hierarchical structure. More recently, Kennes and Smets [1990] apply fast Möbius transforms to reduce redundant computations and thus improve the efficiency even for the general case.

In this paper, we are concerned with the efficient combination for more than two belief functions. We use as a basis existing methods for combining two belief functions: a direct combination based on Dempster's rule and an indirect combination through Möbius transforms. We further explore parallel algorithms for combining more than two belief functions in order to improve the efficiency, as different pieces of evidence can be combined in any order as long as they are independent of each other.

To further test our algorithms, we consider a medical domain that involves the diagnosis of different types of canine liver diseases (McLeish et al. [1989], [1990], [1991]). This is a domain on which doctors have difficulty predicting precise or single outcomes, as both the numbers of possible outcomes (14) and available tests (40) are quite large. In terms of the DS-theory, this would require a combination of 40 belief functions over a frame of 14 different hypotheses[1]. Although our parallel algorithms can largely speed up the implementation, the amount of time used is still quite long. Fortunately, for practical applications, especially our domain, we found that most of the subsets have zero probabilities; the number of subsets that have non-zero probabilities, called the focal elements, are just about 10 on average. Thus, special versions of our algorithms can be designed to facilitate the practical application. Our algorithms are all implemented on a Sequent machine using the parallel C language and the experimental results are reported later in detail.

2 Review of the DS-theory

In DS-theory, probabilities are assigned directly to subsets of a frame of hypotheses, called a mass function (m). Two pieces of evidences can be combined using the Dempster's rule, where m_1 and m_2 are the mass functions for the given evidences:

$$m(B) = \frac{\sum\{m_1(B_1)m_2(B_2) \mid B_1 \cap B_2 = B\}}{\sum\{m_1(B_1)m_2(B_2) \mid B_1 \cap B_2 \neq \emptyset\}}$$

The rule, as stated in [Buchanan and Shortliffe, 1984], provides a way of narrowing the hypothesis set with the

*This research has been supported by the NSERC Networks of Centers of Excellence Program in Canada.

[1] See [McLeish and Song, 1991] for the general framework of our expert system for diagnosing canine liver diseases.

accumulation of evidence and naturally captures the process of diagnostic reasoning in medicine and expert reasoning in general.

There are two ways for combining mass functions proposed in the current literature ([Shafer, 1976] and [Kennes and Smets, 1990]). One is the direct combination based on the Demspter's rule, for which it can be shown that the following theorem holds:

Theorem 2.1 *The direct implementation of the Dempster's rule needs* $(2^n-1)^2$ *additions and* $2^n(2^n-1)$ *multiplications.*

The other way for combining mass functions is the indirect combination through Möbius transforms. Based on a mass function, a commonality function (Q) is further defined in [Shafer, 1976]:

$$Q(A) = \sum\{m(B) \mid B \supseteq A\}$$

With commonality functions, the combination of different evidences is reduced to the multiplication of the commonality functions,

$$Q(A) = KQ_1(A)\ldots Q_n(A)$$

where K^{-1} is a constant that does not depend on A.

A Möbius transform is a function defined over a partially ordered set. For example, the computations from m to Q and vice versa are all Möbius transforms. The idea of a fast Möbius transform is to decompose the whole transform into a series of simple transforms [Kennes and Smets, 1990]. In each step, as illustrated in figure 1, we only consider one hypothesis and its related transform. For example, the first step will achieve the transform: $\{(X,Y) \mid X \neq \emptyset$ and $(Y = X$ or $Y = X \cup \{c\})\}$, where X and Y are two subsets of Θ. Then, by recursively doing this for all the hypotheses, we will be able to transform from one function to another function.

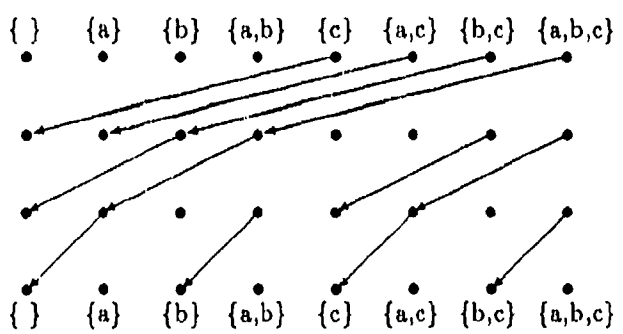

Figure 1: Diagram for the Transform: $m \rightarrow Q$

To combine mass functions, we follow the path from $\{m_i\}$ to $\{Q_i\}$ to Q to m, as shown in figure 2. However, although the transform from Q to m is not provided in [Shafer, 1976], it can be proved, following a similar approach, that the following lemma holds.

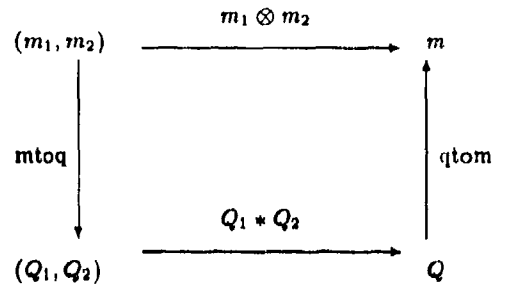

Figure 2: Combination through Möbius Transform

Lemma 2.1 *Suppose* m *and* Q *are two functions defined over a frame* Θ*, then we have:*

$$Q(A) = \sum_{B \supseteq A} m(B) \text{ iff } m(A) = \sum_{B \supseteq A} (-1)^{|B-A|} Q(B).$$

Based on the above lemma, we can now construct a fast Möbius transform from Q to m. It is the same as the transform from m to Q except that all the links have weighting factor (-1) (see [Kennes and Smets, 1990] for detailed discussions).

Theorem 2.2 *The indirect implementation of Dempster's rule through Möbius transforms needs* $3n2^{n-1}$ *additions and* 2^{n+1} *multiplications.*

3 Algorithms for Combining Belief Functions

In this section, we consider how to combine r pieces of evidence efficiently, with $r > 2$. In particular, we present three pairs of algorithms for combining r mass functions: sequential, parallel, and practical methods.

3.1 Sequential Combination Methods

Based on the two methods introduced earlier, we can provide two sequential algorithms for combining more than two belief functions. A sequential algorithm based on Dempster's rule can be given as follows:

algorithm 3.1 sequential & direct implementation
input $m[1:r][0:2^n-1]$, r bodies of mass functions, and n, the cardinality of the frame
output $m[1][0:2^n-1]$, the combined mass function
begin
  for $i = 2$ **step** 1 **until** r **do**
    comb-two($m[1], m[i]$)
  endfor
end

Here, we use a n-digit binary number to represent a frame of size n, and for each subset, the ith element is 1 if the corresponding element is in the subset. Also, "comb-two" is a procedure for combining two mass functions.

Corolary 3.1 *Algorithm 3.1 needs* $(r-1)(2^n-1)^2$ *additions and* $(r-1)2^n(2^n-1)$ *multiplications.*

Another way of implementing the Dempster's rule is to compute the combined mass function indirectly through Möbius transforms. A sequential algorithm for this method can be given as follows:

algorithm 3.2 sequential & indirect implementation
begin
  for $i = 1$ **step** 1 **until** r **do**
    mtoq($m[i]$)
  endfor

  for $i = 0$ **step** 1 **until** $2^n - 1$ **do**
    for $j = 2$ **step** 1 **until** r **do**
      $m[1][i] \leftarrow m[1][i] * m[j][i]$
    endfor
  endfor

  qtom($m[1]$)
end

Corolary 3.2 *Algorithm 3.2 needs* $n(r+1)2^{n-1}$ *additions and* $r2^n$ *multiplications.*

3.2 Parallel Combination Methods

Since in DS-theory, different pieces of evidence can be combined in any order as long as they are independent of each other, we can further explore parallel algorithms for the combination of more than two belief functions.

algorithm 3.3 parallel & direct implementation
begin
  while $r > 1$ **do**
    $r' = r/2$
    for $i = 1$ **step** 1 **until** r' **do in parallel**
      comb-two($m[i], m[r'+i]$)
    endfor
    if odd(r) **then**
      $m[r'+1] = m[r]; r = r'+1$
    else $r = r'$
  endwhile
end

Corolary 3.3 *Algorithm 3.3 needs* $\lceil \log r \rceil (2^n-1)^2$ *additions and* $\lceil \log r \rceil 2^n(2^n-1)$ *multiplications, where* $\lceil \log r \rceil$ *stands for the smallest integer that is greater or equal to* $\log r$.

algorithm 3.4 parallel & indirect implementation
begin
  for $i = 1$ **step** 1 **until** r **do in parallel**
    mtoq($m[i]$)
  endfor

  for $i = 0$ **step** 1 **until** $2^n - 1$ **do in parallel**
    for $j = 2$ **step** 1 **until** r **do**
      $m[1][i] \leftarrow m[1][i] * m[j][i]$
    endfor
  endfor

  qtom($m[1]$)
end

Corolary 3.4 *Algorithm 3.4 needs* $n2^n$ *additions and* $2^n + r$ *multiplications.*

3.3 Practical Combination Methods

To further test our algorithms, we choose a medical domain that involves the diagnosis of canine liver diseases. We found that for such a domain, most of the mass functions only have a small number of non-zero subsets, or focal elements. Although the above algorithms work for general cases, for practical reasons, we must revise them to facilitate the almost null distribution of mass functions.

In the following we first provide a revised procedure for direct combination based on Dempster's rule.

function comb-two'(m_1, m_2, L_1, L_2)
begin
  for $i = 1$ **step** 1 **until** L_1 **do**
    for $j = 1$ **step** 1 **until** L_2 **do**
      $s \leftarrow s_1[i]$ & $s_2[j]$
      $m[s] \leftarrow m[s] + m_1[i] * m_2[j]$
    endfor
  endfor
  $K \leftarrow 1 - m[0]$
  for $i = 1$ **step** 1 **until** $2^n - 1$ **do**
    if $m[i] > 0$ **then**
      $L \leftarrow L + 1$
      $s_1[L] \leftarrow i$; $m_1[L] \leftarrow m[i]/K$
    endif
  endfor
  return L
end

Here, "&" is the bitwise operator for the logical operation "AND", corresponding to the intersection operation between two subsets.

Then a parallel algorithm for combining more than two mass functions can be designed as follows:

algorithm 3.5 practical par. & dir. implementation
begin
  while $r > 1$ **do**
    $\leftarrow r/2$
  ...r $i = 1$ **step** 1 **until** r' **do in parallel**
    $L[i] \leftarrow$ comb-two'($m[i], m[r'+i], L[i], L[r'+i]$)
  endfor
  if odd(r) **then**
    $m[r'+1] \leftarrow m[r]; r \leftarrow r'+1$
  else $r \leftarrow r'$
  endwhile
end

To see how speed can be gained for the above algorithm, let us consider our domain of canine liver diseases. For a frame of size 14, 2^{14} gives us 16,384. Thus, the direct combination of two mass functions would require $(2^{14}-1)^2$ additions and $2^{14}(2^{14}-1)$ multiplications.

However, the above improved direct combination would only need about 100 additions and 110 multiplications, as the average number of focal elements is 10 for any mass functions in our domain (see [McLeish and Song, 1991] for different methods of extracting mass functions from medical data collected over time).

Similarly, we can add a testing statement in a Möbius transform and only perform an addition when the new element is non-zero. Since the cost of a testing statement is usually less than an arithmetic operation, we would expect some saving of time when most of the subsets have zero probabilities. The modified algorithm based on the Möbius transforms will be called algorithm 3.6 in our experiments.

4 Experimental Results

Our algorithms are all implemented on a Sequent Symmetry machine using the Parallel C language [Osterhaug, 1989]. A Sequent machine has an architecture of truly multiple processors and a shared memory, all connected through a system bus. This provides a way for increasing the accessibility of data and minimizing the communication cost. As a result, we can actually run our algorithms on this machine and observe the improvement of speed for a problem of reasonable size.

In our experiments, we run our algorithms on a machine of ten processors. Our results can further be improved when more processors are available, say 16 or 32, which become more and more common for Sequent machines. Although our system is not large, it already shows the potential of using parallel algorithms for efficiently combining belief functions.

# Mass	Alg3.2	Alg3.4	Alg3.5	Alg3.6
02	13.26	9.18	0.43	7.56
03	17.71	9.25	0.95	7.59
04	22.19	9.31	0.95	7.72
05	26.74	9.37	1.51	7.76
08	40.22	13.88	1.49	11.30
10	49.19	14.00	2.04	11.46
15	71.68	18.65	2.35	15.15
16	76.21	23.01	2.34	18.65
20	94.21	23.30	2.88	18.86
25	117.40	27.93	3.52	22.57
30	140.49	32.60	3.54	26.29
32	149.74	37.03	3.86	29.70
35	164.69	37.20	4.39	30.01
40	188.18	41.84	4.39	33.67

Table 1: Results of Sequential and Parallel Experiments

As our results illustrate, for the general case, the parallel implementation based on the fast Möbius transforms (algorithm 3.4) is the most efficient. However, for many real applications where most of the subsets have zero masses, the parallel implementation based on the improved direct combination (algorithm 3.5) is still the most efficient[2].

Further work is being carried out to minimize redundant computations in a Möbius transform and explore parallelism in Dempster's rule. Methods working with continuous data are also being investigated with an application to our domain of liver disease diagnosis.

References

Barnett, J.A. 1981. Computational methods for a mathematical theory of evidence. In *Proceedings of the IJCAI Conference*. 868–875.

Buchanan, B.G. and Shortliffe, E.H. 1984. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley Publishing Company.

Kennes, R. and Smets, P. 1990. Computational aspects of the Möbius transform. In *Proceedings of the Sixth Uncertainty Management Conference*. 344–351.

McLeish, M. and Song, F. 1991. A framework for medical expert systems using Dempster-Shafer theory. Submitted to First World Congress on Expert Systems.

McLeish, M.; Cecile, M.; Yao, P.; and Stirtzinger, T. 1989. Experiments using belief functions and weights of evidence on statistical data and expert opinions. In *Proceedings of the 5th Uncertainty Management Conference*. 253–264.

McLeish, M.; Stirtzinger, T.; and Yao, P. 1990. Using weights of evidence and belief functions in medical diagnosis. In *Proceedings of the AAAI Spring Symposium, AI in Medicine*. 132–136.

McLeish, M.; Yao, P.; and Stirtzinger, T. 1991. Experiments on the use of belief functions for medical expert systems. *Journal of Applied Statistics, Special Issue on Statistics and Expert Systems* 155–174.

Osterhaug, A., editor 1989. *Guide to Parallel Programming on Sequent Computer Systems*. Prentice Hall, second edition.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers.

Shafer, G. and Logan, R. 1987. Implementing Dempster's rule for hierarchical evidence. *Artificial Intelligence* 33:271–298.

Shafer, G.; Shenoy, P.P.; and Mellouli, K. 1987. Propagating belief functions in qualitative Markov trees. *International Journal of Approximate Reasoning* 1:349–400.

Shafer, G. 1976. *A Mathematical Theory of Evidence*. Princeton University Press.

[2]The experiments for algorithms 3.1 and 3.3 are not fully conducted as even for one combination, it already takes about 1.5 hours on our Sequent machine.